Association Rules



เป้าหมาย → หา hidden relations
e.g. คนซื้อผ้าอ้อม ซื้อเบียร์ด้วย

นิยาม: ☆ itemset : กลุ่มของสินค้า เช่น {A, B}
☆ K-item set : Itemset มีสมาชิก k ตัว
☆ Support : ความถี่ที่ itemset ปรากฏมา (ความนิยม)

$$\rightarrow Support(A \cup B) = \frac{Count(A \cup B)}{Total\ Transaction}$$

☆ Confidence : ความน่าจะเป็นที่เกิด A แล้วเกิด B (ความเชื่อมั่น)

$$\rightarrow Confidence(A \rightarrow B) = \frac{Support(A \cup B)}{Support(A)}$$

☆ Large (frequent) Itemset : Itemset
↳ มีค่า support ≥ treshold ที่กำหนดไว้

☆ Lift : บอกว่าการเกิดของ A ส่งผลให้ B เกิด มากกว่าปกติ เท่าไหร่ (ผลเชิงบวก/ลบ)

$$\rightarrow Lift(A \rightarrow B) = \frac{Confidence(A \rightarrow B)}{Support(B)} = P(A \cup B) / P(A) \cdot P(B)$$

Lift > 1 : positive
Lift = 1 : indipendent
Lift < 1 : Negative



Apriori Algorithm

Concept : ใช้การตัดทางเลือก (Pruning) เพื่อลด จ.น. candidates ↓ ที่ต้องตรวจสอบ

Apriori Property (กฎหลัก)

1) Downward Closure : If any itemset is Large, all its subset shall be Large Too
2) If any itemset is not Large, all its subset is not Large

Steps :
1) Counting : นับความถี่ของ 1-itemset ที่ผ่านเกณฑ์ (L_1)
2) Join Step : ตรวจสอบ Candidate C_k ว่ามี subset ขนาด k-1 ไหนบ้างเพื่อสร้าง Candidate C_k



○ เงื่อนไข : สมาชิกตัวหน้า k-2 ตัวต้องเหมือนกัน ตัวสุดท้ายต้องต่างกัน (เรียง น้อย → มาก)

3) Prune Step : Check candidate C_k ว่ามี subset ขนาด k-1 ไหนบ้าง ที่ไม่อยู่ใน L_{k-1} ถ้ามี ให้ตัดทิ้ง
4) วนซ้ำ จนกว่า จะสร้าง candidate ต่อไม่ได้

Frequent Pattern Growth (FP-Growth)

จุดเด่น : Faster than Apriori เพราะ ไม่สร้าง candidate set
Scan database แค่ 2 รอบ

ขั้นตอน :
1) Scan 1 : Count freq. of all, ตัดตัวที่ไม่ผ่านเกณฑ์ทิ้ง
2) Ordering : เรียงลำดับสินค้า high → low freq. (called f-list)
3) Scan 2 (buit tree) : read each row of transaction แล้ววาดลง FP-tree
☆ เรียงสินค้าใน transaction ตาม F-list เสมอ
☆ ถ้าเส้นทางซ้ำให้บวก count เพิ่ม ถ้าไม่ให้แตกกิ่งใหม่
4) Mining (Conditional pattern base)
☆ เริ่มจาก Item ตัวท้ายสุดใน F-list (least freq.)
☆ หาทุกเส้นทาง (Path) จาก root มาถึง node (ไม่รวมตัวเอง) เรียกว่า "conditional pattern base"
☆ สร้าง conditional FP-Tree จาก path เหล่านั้น (นับ freq. ใหม่ และตัดตัวที่ไม่ผ่านเกณฑ์ทิ้ง)
☆ รวมผลลัพธ์ → ได้ freq. pattern

Classification

Decisions Tree

เป็น Flowchart ที่แตกกิ่งตามเงื่อนไข attribute → คำตอบ
→ การเลือก attribute แตกกิ่ง : ต้องเลือกตัวที่ pure ที่สุด

● impurity measure :
1) Entropi (ID3) : $Info(D) = -\sum_{i=1}^{m} P_i \log_2(P_i)$
→ ถ้าข้อมูลแบ่งกลุ่มชัดเจน (pure) Entropi = 0
→ ถ้าปนกันมั่ว (50/50) Entropi = 1
2) Information Gain : ค่า Entropi ที่ลดลง จากการแบ่งด้วย Attribute A (ยิ่งมาก → ดี)
$\rightarrow Gain(A) = Info(D) - Info_A(D)$
3) Gain Ratio : ปรับปรุง gain → แก้ปัญหา attribute ที่มีค่า unique มาก
4) Gain Index : $Gini(D) = 1 - \sum_{i=1}^{m} P_i^2$

Naive Bayes Classifier

หลักการ : ใน bayes prob. theo. โดยสมมติฐาน Independence Assumption" ที่เชื่อว่าทุกตัวอิสระต่อกัน คำนวณง่ายขึ้นมาก

สูตร : หาคลาส C_i ที่ทำให้ค่า $P(C_i) \times \prod P(x_k|C_i)$ สูงสุด
☆ $P(C_i)$: ความน่าจะเป็นของคลาสนั้นๆ (Prior)
☆ $P(x_k|C_i)$: ความน่าจะเป็นของ feature x_k เมื่อกำหนด class C_i

K-nearest Neighbor (KNN)

Concept : ดูคนข้างบ้าน เมื่อมีจุดใหม่เข้ามา ให้หาจุดข้อมูลเดิม nearest k แล้วโหวต (majority vote) → ไม่ต้องสร้าง model ล่วงหน้า

Support Vector Machine

Goal : หาเส้นแบ่ง (hyperplane) ที่แบ่งคลาสดีที่สุด ดูจาก margin (เส้นแบ่ง ←dis→ ข้อมูล)
Concept : need Maximize margin เพื่อให้ model ทนต่อ Noise
Support Vectors : จุดข้อมูลที่อยู่ชิดขอบ margin ที่สุด (ตัวสำคัญในการสร้างเส้นแบ่ง)
Kernel Trick : กรณีแบ่งข้อมูล non-linear ใช้ f(x) Kernel แปลงข้อมูลไปมิติสูง

Clustering : K-means

→ เป็นการ partitioning

Steps :
1) กำหนดค่า k และสุ่มจุด ศูนย์กลาง
2) Assign : นำข้อมูลแต่ละจุด เข้ากลุ่มของ nearest ctr
3) Update : คำนวณ หา new mean of each group แล้วย้าย m to there
4) วน 2) 3) จนกว่าจุด กลางไม่เปลี่ยน ต.น.

⚠ ไวต่อค่า outlier (เพราะใช้ mean) & results may change by จุดเริ่มสุ่ม

Hierarchical Clustering

→ Agglomerative (bottom-up) : เริ่มจากแต่ละจุดเป็นกลุ่มตัวเอง แล้วจับคู่ใกล้กันรวมไปเรื่อยๆ จนเหลือแค่เพียงกลุ่มเดียว
→ Divisive (Top-down) : เริ่มจากก้อนเดียว แล้วค่อยๆ แบ่งย่อย
→ Dendrogram : แผนภาพต้นไม้ แสดงลำดับการรวมกลุ่ม
→ วิธีการ วัดระยะห่าง การรวมกลุ่ม (linkage methods)
1) Single Linkage (min) : วัดจากจุดที่ใกล้ที่สุดของ 2 group
con : may cause "Chain effect" กลุ่มยาวเฟื้อย
2) Complete Linkage (max) : วัดจากจุดที่ไกลกันที่สุด ของ 2 groups
pro : ได้กลุ่ม ทรงกลม แน่น
3) Group Average : average distance of all points
4) Centroid : average distance of 2 groups' จุดศูนย์กลาง
5) Median : ใช้จุดกึ่งกลางเพื่อลดผลกระทบเมื่อกลุ่มขนาดต่างกันมารวมกัน

การจัดระเบียบข้อมูล binary (Monothetic)

→ กรณีข้อมูลเป็น 0/1 เช่น (มี/ไม่มี) ใช้วิธีการแบ่งกลุ่ม โดยพิจารณาทีละ attribute
→ ใช้ Chi-square measure (χ^2) เพื่อวัด relation betw. attribute ก่อนการแบ่งกลุ่ม

$$สูตร : \chi^2 = \frac{(ad-bc)^2 N}{(a+b)(a+c)(b+d)(c+d)}$$